



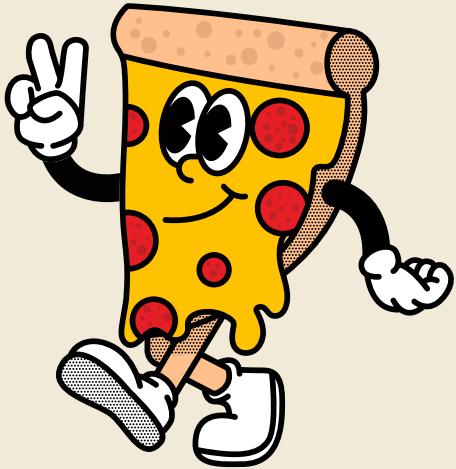
Today's special

PIZZA



ORDER NOW

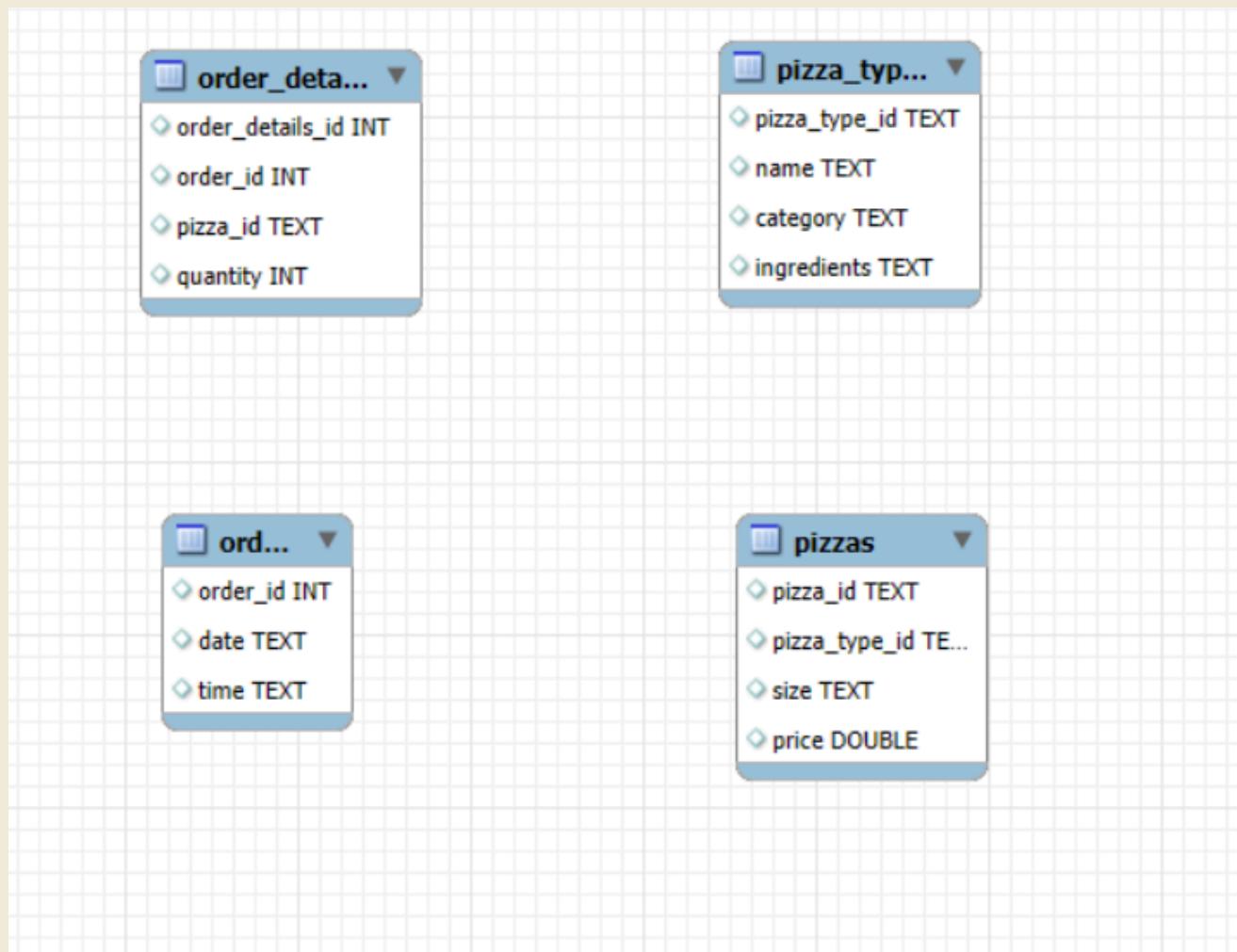
WWW.AYUSH.PIZZA.COM



Hello !

My name is Ayush, and today I'm excited to showcase my SQL project on pizza sales! In a world where data drives business decisions, this project delves into how SQL can unlock insights from a pizza sales database, helping us understand customer behavior, sales trends, and optimize business strategies. From analyzing the most popular toppings to tracking peak sales hours, this project provides a comprehensive look at how data can transform everyday operations into meaningful business growth.

Pizza Schema





Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

Result Grid

	total_orders
▶	21350



Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05



Identify the highest-priced pizza.

```
select name,price from pizzas
join pizza_types on pizzas.pizza_type_id = pizza_types.id
order by price desc
limit 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



List the top 5 most ordered pizza types along with their quantities.

```
select pizza_types.name, sum(order_details.quantity) as quantity from pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by quantity desc limit 5;
```

Result Grid | Filter Rows:

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(time);
```

Result Grid		
	HOUR(time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1



Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT  
    category, COUNT(name) AS names  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid | Filter F

	category	names
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) AS order_per_day
FROM
    (SELECT
        date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS order_quantity;
```

Result Grid		Filter Rows:
	order_per_day	
▶	138	



Determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



Calculate the percentage contribution of each pizza type to total revenue.

```
select pizza_types.category, round(sum(order_details.quantity * pizzas.price) / (select
round(sum(order_details.quantity * pizzas.price),
2) as total_sales
from
order_details
join
pizzas on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

THANK YOU!!